HINFShot: A Challenge Dataset for Few-Shot Node Classification in Heterogeneous Information Network

Zifeng Zhuang
Westlake University
Hangzhou, Zhejiang, P.R. China
Nankai University
Nankai District, Tianjin, P.R. China
dragonzhuang@126.com

Xintao Xiang
The Australian National University
Canberra, Australia
xintao.xiang@anu.edu.au

Siteng Huang
Westlake University
Westlake Institute for Advanced Study
Hangzhou, Zhejiang, P.R. China
huangsiteng@westlake.edu.cn

Donglin Wang*
Westlake University
Westlake Institute for Advanced Study
Hangzhou, Zhejiang, P.R. China
wangdonglin@westlake.edu.cn

ABSTRACT

Few-shot learning aims to generalize to novel classes. It has achieved great success in image and text classification tasks. Inspired by such success, few-shot node classification in homogeneous graph has attracted much attention but few works have begun to study this problem in Heterogeneous Information Network (HIN) so far. We consider few-shot learning in HIN and study a pioneering problem HIN Few-Shot Node Classification (HIN-FSNC) that aims to generalize the node types with sufficient labeled samples to unseen node types with only few-labeled samples. However, existing HIN datasets contain just one labeled node type, which means they cannot meet the setting of unseen node types. To facilitate the investigation of HIN-FSNC, we propose a large-scale academic HIN dataset called *HINFShot*. It contains 1,235,031 nodes with four node types (*author, paper, venue, institution*) and all the nodes regardless of node type are divided into 80 classes. Finally, we conduct extensive experiments on *HINFShot* and the result indicates a significant challenge of identifying novel classes of unseen node types in HIN-FSNC.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Computing methodologies** → **Supervised learning by classification**.

*Corresponding author.
Authors would like to acknowledge funding support from the Westlake University and Bright Dream Joint Institute for Intelligent Robotics.

ICMR '21, August 21–24,2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463614

KEYWORDS

Heterogeneous Information Network, Few-shot Learning, Academic Network Mining

ACM Reference Format:
Zifeng Zhuang, Xintao Xiang, Siteng Huang, and Donglin Wang. 2021. HINFShot: A Challenge Dataset for Few-Shot Node Classification in Heterogeneous Information Network. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21),August 21–24,2021,Taipei, Taiwan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3460426.3463614

1 INTRODUCTION

Few-shot learning (FSL) aims to recognize novel classes from very few labeled examples [23]. Research on FSL has been carried out in many tasks, such as image classification [7, 22, 23, 26], object detection [6] and text classification [30]. Recently, more and more attention has been paid to graph FSL tasks including graph classification [2], link prediction [1] and node classification [4, 11, 15, 29, 33]. Although these existing works have achieved great success, they take only homogeneous graph into consideration while few works study the FSL in Heterogeneous Information Network (HIN).

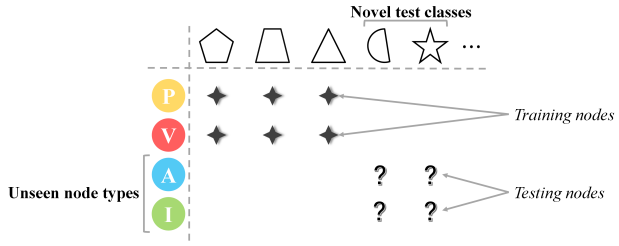


Figure 1: The visualization of HIN-FSNC. In experiments, we use nodes with node types P(aper) and V(enue) to predict A(uthor) and I(nstitute) with few-labeled nodes.

HIN contains different node and edge types that are extremely useful in application. When modeling real systems, we can use different node and edge types to represent various entities and complex relations. In this way, real systems, such as academic, social and biological networks, can be well abstracted by preserving more details compared to homogeneous graph. Generally, the distribution of labeled samples is uneven among different node and edge types. One important and meaningful problem is to predict the labels of nodes with few-labeled node types using the sufficient information from other nodes with different node types. Since the label spaces of different node types are not the same, novel classes must exist. We summarize it as HIN Few-Shot Node Classification (HIN-FSNC) showed in Figure 1, a pioneering problem introducing unseen node type in FSL problem under the HIN scenario.

Table 1: The comparison between classic HIN datasets and *HINFShot*. Here N/A means one node type does not exist or has no label. It can be observed that classic datasets only have one labeled node type.

Dataset	Statistics	Node Type			
DBLP		Author	Paper	Venue	Term
v1 [8, 13]	nodes	4,057	14,328	20	7,723
	classes	4	N/A	N/A	N/A
v2 [19, 27]	nodes	4,057	14,328	20	8,789
	classes	4	N/A	N/A	N/A
v3 [18]	nodes	1,960	7,907	N/A	1,975
	classes	N/A	4	N/A	N/A
v4 [32]	nodes	67,950	70,910	97	N/A
	classes	N/A	4	N/A	N/A
IMDB		Movie	Director	Actor	Keyword
v1 [8, 13]	nodes	4,728	2,081	5,257	N/A
	classes	3	N/A	N/A	N/A
v2 [27]	nodes	4,780	2,269	5,841	N/A
	classes	3	N/A	N/A	N/A
v3 [19]	nodes	4,275	2,082	5,431	7,313
	classes	3	N/A	N/A	N/A
v4 [18]	nodes	3550	1,762	4,441	N/A
	classes	3	N/A	N/A	N/A
		Author	Paper	Venue	Institution
HINFShot	nodes	321,237	889,431	15,743	8,620
	classes	80	80	80	80

The details of representative HIN datasets for node classification are summarized in Table 1 and both DBLP and IMDB contains four different versions. All the datasets are infeasible to be used to study HIN-FSNC due to one dominant limitation that only one node type has labels, which means it is impossible to have unseen node types during the test phase. For instance, the DBLP (v1) [8, 13] includes four node types in which only node type *author* has four classes. So we can only study *author* classification and it is impossible to investigate the problem of HIN-FSNC. Therefore, we propose a novel dataset for HIN-FSNC problem called *HINFShot* that overcomes the limitation.

HINFShot collects more than one million data items with four node types (*author, paper, venue, institution*) from a billion-scale academic graph, Open Academic Graph [21, 24]. We categorize all the nodes regardless of node type into 80 classes according to periodical partition table [17]. Then we divide these 80 classes into train, validation, novel test classes and then split *HINFShot* based on the divided classes. The feature of each node is composed of two parts: semantic part generated by XLNet [28] and structure part from Metapath2vec [5]. To conclude, *HINFShot* is an academic dataset containing 1,235,031 nodes with 4 node types, 80 classes and 896 dimensional features.

Finally, we conduct extensive experiments of HIN-FSNC on *HINFShot*. The results (Table 3) show that neither optimization-based methods nor metric-based methods can solve HIN-FSNC well. We believe that the huge distribution gap between different node types cause low performances. It is necessary to design few-shot algorithms considering unseen node types to deal with it. In summary, the main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to pay close attention to FSL in HIN and investigate HIN-FSNC, a challenging problem with both novel classes and unseen node types.
- We propose a large-scale dataset *HINFShot* that contains 4 node types, 1,235,031 nodes and 80 classes for all node types to study HIN-FSNC.
- Extensive experiments are conducted to evaluate the performances of existing methods on *HINFShot*. Detailed analysis indicates a significant challenge of HIN-FSNC.

2 RELATED WORK

2.1 HIN Representation Learning

Early shallow models explore HIN by encoding nodes into embeddings for downstream tasks. [5] introduces meta-path based random walk and achieves great performance. With the development of GNN, some networks dedicated for HIN have been proposed. [27] implicitly changes HIN into homogeneous graph via meta-path and then applies GAT [25] to solve the problem. [13] applies GCN [14] in HIN via implicitly utilizing attention and meta-paths. [12] explicitly homogenizes HIN into homogeneous graph using parallel composite edge (PCE) which is a group of meta-path and then uses GAT [25]. Though these meta-path based methods achieve good performance, they need expert knowledge to design the meta-path. Therefore, methods without manually designed meta-path are developing. [31] designs HetGNN based on random walk to capture both structure and content heterogeneity. [20] attempts to generate meta-path by neural networks automatically. [9] directly encodes structural information of HIN without meta-path. [10] proposes heterogeneous graph transformer (HGT) which uses attention mechanism and does not require manually designed meta-paths as input.

2.2 Graph Few-shot Learning

Various studies have integrated FSL with graph and applied to graph classification, link prediction and node classification. For graph

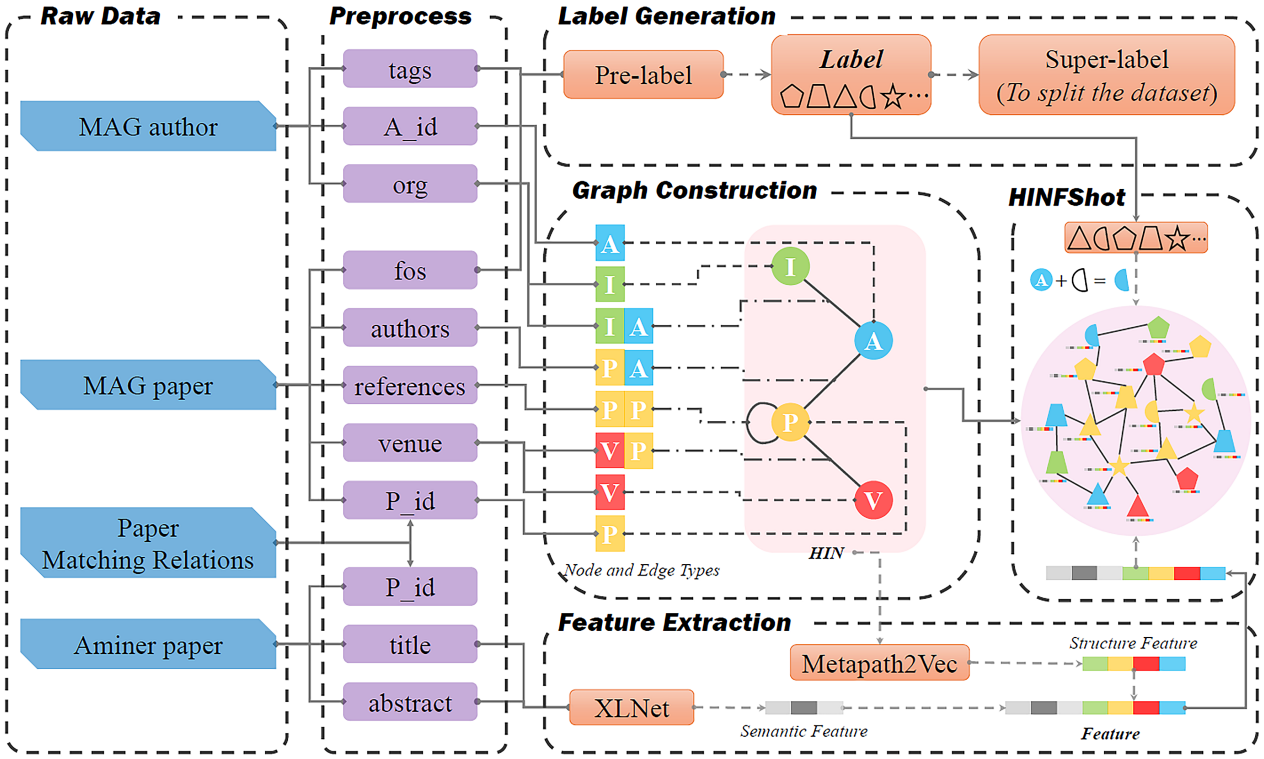


Figure 2: Data Collection and Preparation of *HINFShot*. Colors in Graph Construction represent node types and shapes in HINFShot represent labels in figure. So a blue semi-circle means an author with one certain label.

classification, [2] utilizes spectral measures to generate super-classes and super-graph for modeling the latent relations between classes. For link prediction, [1] introduces a new framework among multiple graphs and a series of benchmarks. For node classification, [33] simply fuses GCN [14] and MAML [7] to tackle node classification while [29] uses knowledge transfer between graphs. [15] studies node classification by designing a transformation function that captures the multifaceted relationships between nodes on a dataset with only structure but not feature. Furthermore, [11] presents three fundamentally meta-learning problems for both node classification and link prediction. [4] proposes a framework Graph Prototypical Networks (GPN) for few-shot node classification on attributed networks. However, these works all focus on homogeneous graph while we consider FSL in HIN.

3 PRELIMINARY

Heterogeneous Information Network. Heterogeneous information network is defined as $G = (V, E, \mathcal{A}, \mathcal{R})$ where V is the node set and E is the edge set. Each nodes $v \in V$ and each edges $e \in E$ are associated with their type mapping functions $\tau(v) : V \rightarrow \mathcal{A}$ and $\phi(e) : E \rightarrow \mathcal{R}$, respectively.

HIN Few-Shot Node Classification (HIN-FSNC). We formulize the definition of HIN-FSNC as below. We use $C = \{c_1, c_2, \cdots, c_n\}$ to denote all the classes and $\mathcal{A} = \{a_1, a_2, \cdots, a_m\}$ to denote all the node types in HIN. V_{tr} denotes the train set that has abundant samples where for $\forall v \in V_{tr}$, $\tau(v) \in \{a_1, a_2, \cdots, a_j\}$ and the label $c_{tr} \in C_{tr}$. V_{ts} is the test node set with $\forall v^* \in V_{ts}$, $\tau(v^*) \in \{a_{j+1}, a_{j+2}, \cdots, a_m\}$ and label $c_{ts} \in C_{ts}$. V_{ts} has only few-labeled samples. Note that $C_{tr} \cap C_{ts} = \emptyset$. HIN-FSNC predicts all the labels of v^* using v under the setting of FSL.

Motivation. Current existing datasets cannot be used to study HIN-FSNC. Table 1 shows that all the existing datasets are impossible to investigate unseen node type since only one node type is labeled. Motivated by this situation, we propose a large-scale HIN benchmark especially for HIN-FSNC called *HINFShot*. It could be a standard dataset to be widely used in the future for exploring few-shot node classification in HIN.

4 DATASET

This whole process contains five steps and each step corresponds to the component in Figure 2 one by one.

4.1 Raw data

Our *HINFShot* is created from Open Academic Graph (OAG), a large knowledge graph unifying two billion-scale academic graphs: Microsoft Academic Graph (MAG) [21] and AMiner [24]. OAG contains three types of raw data, *papers*, *authors* and *venues* that may come from MAG or AMiner. Because one *paper*, *author* or *venue* may appear in both graphs, matching relations between them are generated. We utilize raw data *paper* and *author* from MAG to form

our dataset. Note that as MAG *paper* has no abstract, we use the corresponding *paper* in Aminer to replace it by *paper* matching relations.

4.2 Preprocess

OAG contains abundant entities but the qualities of them vary widely. Therefore, we abandon lots of nodes to guarantee that our proposed dataset *HINFShot* is of high quality. All the following steps strictly adhere to the principle "Fewer but better" and fortunately, the billion-scale raw dataset supports this strategy.

For MAG *author*, data items with "id", "org" and "tags" are retained, where "id" is the unique identification of each *author*, "org" represents the institution where the *authors* work, and "tags" is the research directions that the *author* is interested in and will be used to generate the pre-label.

MAG *paper* items with "id", "venue", "authors", "references" and "fos" are retained, where similarly, "id" is the unique identification, "venue" is where the *paper* is published, "authors" are the authors of this *paper*, "references" are the cited papers, and "fos" contains some related fields with weights. The pre-label of *papers* will be generated from "fos". For Aminer *paper*, we only preserve "id", "title" and "abstract". "id" is used to match the *paper* in MAG by *paper* matching relations. "title" and "abstract" will be fed into a word embedding model to generate node features.

4.3 Graph construction

After preprocessing, all the node types and edge types are available. For node types, "id" of *authors* denotes **A**, "org" is the institute of **A** and denotes node type **I**, and "id" of *papers* denotes **P** where its "venue" is another node type **V**.

For edge types, **I** is where **A** works and denotes edge type **AI**. **V** denotes where **P** is published and indicates edge type **PV**. **A** denotes authors of **P** and represents edge type **PA**. **P** can be reference of another **P** and represent edge type **PP**. Then, the structure of HIN is constructed based on the stated information above.

4.4 Label generation

HIN-FSNC requires that nodes from arbitrary node type have labels. "tags" indicates many interested directions of **A** and we choose the first one to be the pre-label. We choose **P**'s related field with maximum weight as the pre-label of **P**. Then, we get 226,381 pre-labels but most of them appear rarely. Only nodes with pre-labels of high frequency are retained to guarantee the quality and avoid noise while others are abandoned.

Periodical partition table [17] categorizes academic journals into nearly 200 sub-disciplines and 18 major disciplines. Since the pre-labels are related to these academic journals, we manually categorize pre-labels into these sub-disciplines and view them as labels. Some pre-labels are ambiguous so they are abandoned. The pre-label is retained when and only when we are absolutely certain that one pre-label belongs to which sub-discipline. For node type **V**, we first count the frequencies of its **P** neighbors' labels. Then the label with highest frequency is chosen to be the label of **V**. In this way, we also avoid the phenomenon of multi-label. As for **I**, the label is determined by the same strategy using node type **A**. Under strict screening, only 182 pre-labels with high quality are preserved and categorized into 80 sub-disciplines to be 80 labels.

Furthermore, these 80 labels, namely, sub-disciplines belong 12 major disciplines directly based on Periodical partition table [17]. Here, we view major disciplines as *super-labels*. We divide all these 80 classes according to *super-labels* and then split the data based on the divided classes. Training classes include 8 super-labels which contain 60 classes in total while validation and novel test classes include 2 super-labels which contain 10 classes respectively. Detailed description about the labels of *HINFShot* is presented in supplementary materials.

Table 2: Statistics of *HINFShot*.

#Nodes	1,235,031
#Papers	889,431
#Authors	321,237
#Venue	15,743
#Institution	8,620
#Edges	3,809,724
#Average Node Degree	6.1
#Labels	80
#Split	60/10/10

4.5 Feature extraction

Similar to [10], we concatenate semantic features and structure features to get the features of all nodes. The semantic features of **P** come from the preprocessed "title" and "abstract". We use word embedding model XLNet [28] to process "title" and "abstract" to get semantic features of 768 dimensions. For **A**, **I** and **V**, we average their neighbors' semantic features to get theirs. The structure features come from Metapath2vec [5], a powerful HIN embedding tool that encodes each node into a vector using structure information. We utilize four meta-paths (IAPPPAI, VPAPAPV, APPVPPA, PPAIAPP) to get 128 dimensional structure features. Finally, we concatenate them to obtain 896 dimensional features. The basic statistics is summarized in Table 2.

5 METHODOLOGY

In this section, we present our FSL framework on HIN. An overview of this framework is in Figure 3.

5.1 Task Sampling

Our formulation of task is consistent with episodic training in general FSL. We sample M tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ at each training step and each task $\mathcal{T}_i$ contains support set $\mathcal{S}$ with N classes K samples and query set $\mathcal{Q}$ with N classes K^* samples.

Node sampling. When sampling a task $\mathcal{T}_i$, we first randomly select N different labels $C_{\mathcal{T}_i} = \{c_{i1}, \cdots, c_{iN}\}$. For each class $c_j \in C_{\mathcal{T}_i}$, we randomly sample nodes of specific node types (**P**, **V** for training, **A**, **I** for validation and testing in this paper) to form support set $\mathcal{S}_{ij} = \{(v_k, c_{ij}), k = 1, \cdots, K\}$ and query set $\mathcal{Q}_{ij} = \{(v_k^*, c_{ij}), k = 1, \cdots, K^*\}$.

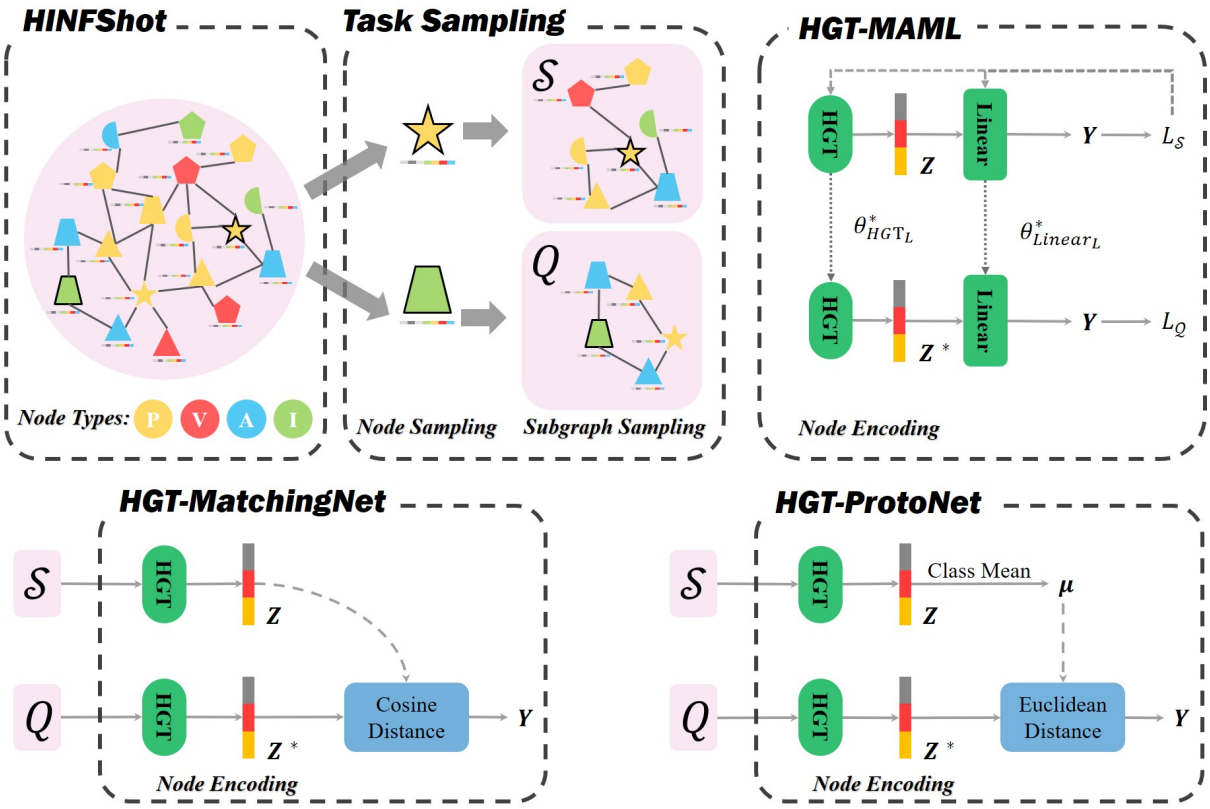


Figure 3: An overview of our methodology which contains two parts, task sampling and HIN-FSL.

Algorithm 1: HGT-MAML

Input: HIN $G = (V, E, \mathcal{A}, \mathcal{R})$; Task learning rate α; Meta learning rate β; Class number N; Task number M;

Initialize the parameters of $\mathrm{HGT}_{\theta}^{(h)}$;
while *not done* **do**
  Sample tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ by section 5.1;
  for $\mathcal{T}_j \in \{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ **do**
    Sample class set $C_{\mathcal{T}_i} = \{c_{i1}, \cdots, c_{iN}\}$;
    for $c_j \in C_{\mathcal{T}_i}$ **do**
      Sample the support and query set $\mathcal{S}_{ij}, \mathcal{Q}_{ij}$;
      Sample local subgraph $G_v, \forall v \in \mathcal{S}_{ij}, \mathcal{Q}_{ij}$;
    end
  end
  for $\mathcal{T}_i \in \{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ **do**
    Evaluate $\nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{S}} \left(\mathrm{HGT}_{\theta}^{(h)}\right)$ by support set;
    Compute $\theta_i' = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}^{\mathcal{S}} \left(\mathrm{HGT}_{\theta}^{(h)}\right)$;
  end
  Update $\theta = \theta - \beta \nabla_{\theta} \sum_i L_{\mathcal{T}_i}^{\mathcal{Q}} \left(\mathrm{HGT}_{\theta_i'}^{(h)}\right)$ by query set;
end

Local subgraph sampling. For each node v, we sample part of its h-hop neighbors to form a local subgraph G_v. We use similar sampling method introduced in [10] that considers the node types and the normalized degree of each node to sample the subgraph. Then the subgraph is used to represent all the available information of the center node v. In the way, we improve computing efficiency significantly by avoiding propagating in the whole HIN.

5.2 HIN-FSL

Given a task $\mathcal{T}_i$ with a series of center nodes and their subgraphs, we use modified HGT [10] as feature extractors to get their embeddings. Then embeddings are fed into MAML [7], ProtoNet [22] or MatchingNet [26] for classification.

Node encoding. Given a center node v with its subgraph G_v, the encoding of HGT can be described as follows:

$$\mathrm{Z}_v = \mathrm{HGT}_{\theta}^{(h)}(v, G_v),$$

where Z_v is the final embedding of center node v and $\mathrm{HGT}_{\theta}^{(h)}$ is HGT with h-layer for h-hop neighbors. Here HGT is modified in order to fit our dataset. Original HGT uses multi-head attention grounded by meta-relations $\langle \tau(s), \phi(e), \tau(t) \rangle$ where s, e and t denote source node type, relation type and target node type, respectively. Since we only consider node types in our dataset, the parameters for relation types are ignored. So it can be represented as

$\langle \tau(s), \tau(t) \rangle$. So the weight matrix in mutual attention $W^{ATT}_{\phi(e)}$ is replaced by $W^{ATT}_{(st)}$. The original expression means the weight matrix is determined by relation type $\phi(e)$ while now it is determined by source and target node type $\tau(s), \tau(t)$. Similarly, the weight matrix in message passing $W^{MSG}_{\phi(e)}$ is replaced by $W^{MSG}_{(st)}$. The prior weight tensor μ is omitted for simplification. For details of HGT, please refer to [10].

HIN-FSL. Then the embeddings can be fed into MAML for few-shot node classification. We change the model f_θ of MAML into $\mathrm{HGT}^{(h)}_\theta$ then we get **HGT-MAML**. The algorithm is shown in Algorithm 1. MAML can be replaced by metric-based methods [22, 26] to get **HGT-ProtoNet** and **HGT-MatchingNet**. In **HGT-ProtoNet** (Algorithm 2), the prototype is calculated by the mean of samples in support set.Then we classify the sample in query set based on its Euclidean distance to every prototype. In **HGT-MatchingNet** (Algorithm 3), for each query sample, we compare its average cosine distance to every support sample in each class. All of them are displayed in Figure 3.

Algorithm 2: HGT-ProtoNet

Input: HIN $G = (V, E, \mathcal{A}, \mathcal{R})$; Learning rate α; Class number N; Task number M; Train shot K;

Initialize the parameters of $\mathrm{HGT}^{(h)}_\theta$;
while *not done* **do**
  Sample tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ by section 5.1;
  for $\mathcal{T}_j \in \{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ **do**
    Sample class set $C_{\mathcal{T}_i} = \{c_{i1}, \cdots, c_{iN}\}$;
    for $c_j \in C_{\mathcal{T}_i}$ **do**
      Sample the support and query set $\mathcal{S}_{ij}, \mathcal{Q}_{ij}$;
      Sample local subgraph $G_v, \forall v \in \mathcal{S}_{ij}, \mathcal{Q}_{ij}$;
    end
  end
  Initialize loss $\mathcal{L} = 0$;
  for $\mathcal{T}_j \in \{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ **do**
    for $c_j \in C_{\mathcal{T}_i}$ **do**
      for $\left(v, c_{ij}\right) \in \mathcal{S}_{ij}$ **do**
        Evaluate Loss $\mathcal{L} = \mathcal{L} + \mathcal{L}_v\left(d, c_{ij}\right)$;
      end
    end
  end
  Update $\theta = \theta - \alpha \nabla_\theta \mathcal{L}$;
end

6 EXPERIMENTS

The experiments are conducted to evaluate the performances of various methods on *HINFShot*.

6.1 Baselines

- Pre-train methods: **FS-HGT**, **FS-GCN** and **FS-GAT** use feature extractors HGT, GCN, and GAT, respectively. They make 60-way classification on the train set and adapt a new linear classifier on the test set While **FS-HGT***, **FS-GCN*** and **FS-GAT*** use cosine distance [3] in testing phase.
- Optimization-based FSL algorithms: **HGT-MAML** has been introduced in section 5.2. **GCN-MAML** and **GAT-MAML** replace HGT [10] with GCN [14] and GAT [25], respectively. **G-Meta** [11] is an optimization-based method that uses GCN as feature extractor and classifies nodes by prototypes.
- Metric-based FSL algorithms: **HGT-ProtoNet**, **GCN-ProtoNet** and **GAT-ProtoNet** simply replace MAML with ProtoNet [22] while **HGT-MatchingNet**, **GCN-MatchingNet** and **GAT-MatchingNet** replace MAML with MatchingNet [26].

Algorithm 3: HGT-MatchingNet

Input: HIN $G = (V, E, \mathcal{A}, \mathcal{R})$; Learning rate α; Class number N; Task number M; Test shot K^*;

Initialize the parameters of $\mathrm{HGT}^{(h)}_\theta$;
while *not done* **do**
  Sample tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ by section 5.1;
  for $\mathcal{T}_j \in \{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ **do**
    Sample class set $C_{\mathcal{T}_i} = \{c_{i1}, \cdots, c_{iN}\}$;
    for $c_j \in C_{\mathcal{T}_i}$ **do**
      Sample the support and query set $\mathcal{S}_{ij}, \mathcal{Q}_{ij}$;
      Sample local subgraph $G_v, \forall v \in \mathcal{S}_{ij}, \mathcal{Q}_{ij}$;
    end
  end
  Initialize loss $\mathcal{L} = 0$;
  for $\mathcal{T}_j \in \{\mathcal{T}_1, \cdots, \mathcal{T}_M\}$ **do**
    for $\left(v^*, c_{ij}\right) \in \mathcal{Q}_{ij}$ **do**
      for $c_j \in C_{\mathcal{T}_i}$ **do**
        Compute average cosine distance by $d_j = \frac{1}{K^*} \sum_v d\left(\mathrm{HGT}^{(h)}_\theta(v), \mathrm{HGT}^{(h)}_\theta(v^*)\right)$, where $(v, c_i j) \in \mathcal{S}_{ij}$;
      end
      Classify v to the closest class;
      Evaluate Loss $\mathcal{L} = \mathcal{L} + \mathcal{L}_v$;
    end
  end
  Update $\theta = \theta - \alpha \nabla_\theta \mathcal{L}$;
end

6.2 Details on Implementation

To make a fair comparison of different feature extractors, same to [10], the node features are encoded by a **node-type aware** linear projection first for mapping data from heterogeneous distributions into same distributions. The query set size of a class is 15. We set sample depth and layer number h to 2 as a trade-off between performance and efficiency. Sampling width is set to 7 which means for each node type we will sample at most 7 nodes in the same depth. All models have same hidden dimensions 280. The details of learning rates are displayed in supplementary materials. Sometimes, the graph is sparse where nodes may only have 1 or 2 neighbors. To overcome it, we use a warm-up strategy. At the start of the training, we filter out the subgraphs whose node numbers are less than

Table 3: Comparisons of few-shot node classification results on *HINFShot*. All results are averaged over 600 episodes and ± denotes 95% confidence intervals over tasks.

Method	5-way Accuracy (%)		
	1-shot	5-shot	10-shot
FS-HGT	30.44± 0.20	39.94 ± 0.22	44.80 ± 0.21
FS-HGT*	30.27 ± 0.20	40.01 ± 0.21	44.78± 0.19
FS-GCN	30.44 ± 0.24	38.77 ± 0.24	41.70± 0.22
FS-GCN*	31.10 ± 0.21	40.33 ± 0.21	44.60 ± 0.20
FS-GAT	29.22±0.23	37.87 ± 0.20	41.26± 0.21
FS-GAT*	30.13± 0.24	38.79 ± 0.19	42.85± 0.21
G-Meta	29.79 ± 0.15	38.45± 0.14	43.75 ± 0.20
HGT-MAML	32.06 ± 0.17	39.67 ± 0.24	42.63 ± 0.19
GCN-MAML	31.18 ± 0.16	37.28 ± 0.19	41.85 ± 0.16
GAT-MAML	31.40± 0.15	39.43± 0.16	44.07 ± 0.19
HGT-ProtoNet	29.54 ± 0.21	37.04 ± 0.22	39.40 ± 0.20
GCN-ProtoNet	30.20 ± 0.20	37.60 ± 0.20	43.31 ± 0.20
GAT-ProtoNet	29.49 ± 0.19	39.40 ± 0.21	43.46 ± 0.20
HGT-MatchingNet	29.17 ± 0.23	35.13 ± 0.23	38.45 ± 0.19
GCN-MatchingNet	31.88 ± 0.23	36.84 ± 0.20	40.14 ± 0.18
GAT-MatchingNet	30.35 ± 0.22	37.45 ± 0.20	41.28 ± 0.19

a preset threshold. We then gradually decrease the threshold with the training forwarding until the threshold becomes 0. This strategy can help the model to converge faster at the beginning of the training.

6.3 Details about Hyperparameters

In this section, we give details about the hyperparameters of experiments. For attention based models HGT and GAT, we fix attention heads to 8. For optimization-based methods, the number of train steps is 5 while the number of test steps is 8. We summarize the hyperparameters in Table 4, Table 5 and Table 6.

6.4 Result

In the following, we present and discuss the performances of baselines on *HINFShot*.

Comparison results: Table 3 summarizes the results of all experiments. Because of **node-type aware** linear projections before feature extractors, the performance gaps between heterogeneous method (HGT) and homogeneous ones (GCN, GAT) are not obvious. Besides, all the performances are pretty low. Even in the setting of 5-way-10-shot, the best result only achieves 44.80%. It indicates that classic methods cannot generalize well to out-of-distribution data in HIN.

Visualization: To analyse intuitively, we utilize t-SNE [16] to visualize the embeddings of test nodes and colored the nodes based on their classes in Figure 4. From the figure, we can find out that for all the methods displayed, the nodes from different classes are mixed together and the boundary is blurry, which indicates that the current algorithms cannot deal with HIN-FSNC very well.

Table 4: Hyperparameters of pre-train methods based experiments.

Method	Setting	Training lr	Adapting lr	Steps
FS-GCN	1-shot	0.006	0.006	24
	5-shot	0.006	0.006	24
	10-shot	0.006	0.006	24
FS-GCN*	1-shot	0.006	0.006	24
	5-shot	0.006	0.006	24
	10-shot	0.006	0.006	24
FS-GAT	1-shot	0.006	0.005	20
	5-shot	0.006	0.005	20
	10-shot	0.006	0.005	20
FS-GAT*	1-shot	0.006	0.005	20
	5-shot	0.006	0.005	20
	10-shot	0.006	0.005	20
FS-HGT	1-shot	0.001	0.005	28
	5-shot	0.001	0.005	28
	10-shot	0.001	0.005	28
FS-HGT*	1-shot	0.001	0.005	28
	5-shot	0.001	0.005	28
	10-shot	0.001	0.005	28

Table 5: Hyperparameters of optimization-based experiments.

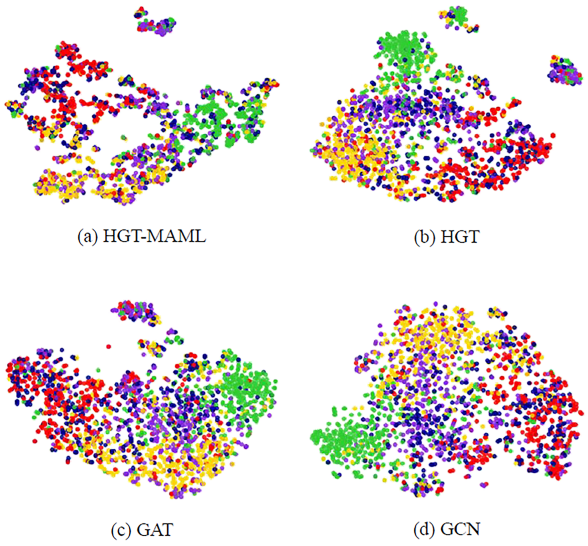
Method	Setting	Inner lr	Update lr
G-Meta	1-shot	0.0060	0.0006
	5-shot	0.0060	0.0006
	10-shot	0.0010	0.0001
GCN-MAML	1-shot	0.0060	0.0006
	5-shot	0.0060	0.0006
	10-shot	0.0060	0.0006
GAT_MAML	1-shot	0.0060	0.0006
	5-shot	0.0060	0.0006
	10-shot	0.0060	0.0006
HGT_MAML	1-shot	0.0200	0.0020
	5-shot	0.0200	0.0020
	10-shot	0.0200	0.0020

7 CONCLUSION AND FUTURE WORK

We introduce a pioneering problem Heterogeneous Information Network Few-Shot Node Classification (HIN-FSNC) that aims to exploit the node types with sufficient labeled nodes and generalize to node types with novel classes and few-labeled nodes. To enable the investigation of HIN-FSNC and overcome the limitation of existing datasets that only one node type has labels, we propose a large-scale academic HIN dataset called *HINFShot*. We conduct

Table 6: Hyperparameters of metric-based expeirments.

Method	Update lr		
	1-shot	5-shot	10-shot
GCN-ProtoNet	0.0008	0.001	0.0008
GAT-ProtoNet	0.001	0.001	0.001
HGT-ProtoNet	0.001	0.002	0.005
GCN-MatchingNet	0.0008	0.0008	0.0008
GAT-MatchingNet	0.0008	0.0008	0.0008
HGT-MatchingNet	0.0008	0.0008	0.002



(a) HGT-MAML (b) HGT

(c) GAT (d) GCN

Figure 4: t-SNE visualization of the test nodes embeddings from the final layers of the model HGT-MAML, HGT, GAT and GCN. It can be found that different classes are not easy to distinguish.

extensive experiments on *HINFShot* and the result indicates a significant challenge of identifying novel classes of unseen node types in the context of HIN-FSNC.

The performances of current methods are not satisfactory due to the fact that significant domain gaps between different node types have been ignored. A promising *future work* is to develop algorithms that can overcome the gaps between different node types and thus improve the accuracy of FSL by the virtue of good generalization.

REFERENCES

[1] Avishek Joey Bose, Ankit Jain, Piero Molino, and William L. Hamilton. 2019. Meta-Graph: Few shot Link Prediction via Meta Learning. *arXiv:1912.09867* (2019).

[2] Jatin Chauhan, Deepak Nathani, and Manohar Kaul. 2020. FEW-SHOT LEARNING ON GRAPHS VIA SUPER-CLASSES BASED ON GRAPH SPECTRAL MEASURES. In *ICLR*.

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A Closer Look at Few-shot Classification. In *ICLR*.

[4] Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. 2020. Graph Prototypical Networks for Few-shot Learning on Attributed Networks.

[5] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*.

[6] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. 2020. Few-shot object detection with attention-RPN and multi-relation detector. In *CVPR*.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

[8] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *WWW*.

[9] Huiting Hong, Hantao Guo, Yucheng Lin, Xiaoqing Yang, Zang Li, and Jieping Ye. 2020. An Attention-Based Graph Neural Network for Heterogeneous Structural Learning.. In *AAAI*.

[10] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW*.

[11] Kexin Huang and Marinka Zitnik. 2020. Graph Meta Learning via Local Subgraphs. *arXiv:2006.07889* (2020).

[12] Tiancheng Huang, Zifeng Zhuang, Shanshan Zhang, and Donglin Wang. 2020. Homogenization with Explicit Semantics Preservation for Heterogeneous Information Network. In *CIKM*.

[13] Di Jin, Zhizhi Yu, Dongxiao He, Carl Yang, Philip S. Yu, and Jiawei Han. 2020. GCN for HIN via Implicit Utilization of Attention and Meta-paths. *arXiv:2007.02643* (2020).

[14] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907* (2016).

[15] Lin Lan, Pinghui Wang, Xuefeng Du, Kaikai Song, Jing Tao, and Xiaohong Guan. 2020. Node Classification on Graphs with Few-Shot Novel Labels via Meta Transformed Network Embedding. *arXiv:2007.02914* (2020).

[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* (2008).

[17] Chinese Academy of Sciences National Science Library. 2021. *Periodical partition table.* http://www.fenqubiao.com/

[18] Chanyoung Park, Donghyun Kim, Jiawei Han, and Hwanjo Yu. 2020. Unsupervised Attributed Multiplex Network Embedding.. In *AAAI*.

[19] Yuxiang Ren, Bo Liu, Chao Huang, Peng Dai, Liefeng Bo, and Jiawei Zhang. 2019. Heterogeneous Deep Graph Infomax. *arXiv:1911.08538* (2019).

[20] Yun Seongjun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo Kim. 2019. Graph Transformer Networks. In *NeurIPS*.

[21] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *WWW*.

[22] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *NIPS*.

[23] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *CVPR*.

[24] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: extraction and mining of academic social networks. In *SIGKDD*.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv:1710.10903* (2017).

[26] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NIPS*.

[27] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*.

[28] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NIPS*.

[29] Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh V Chawla, and Zhenhui Li. 2019. Graph few-shot learning via knowledge transfer. *arXiv:1910.03053* (2019).

[30] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. *arXiv:1805.07513* (2018).

[31] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *SIGKDD*.

[32] Chen Zhang, Guodong Wang, Bin Yu, Yu Xie, and Ke Pan. 2020. Proximity-aware heterogeneous information network embedding. *Knowledge Based Systems* (2020).

[33] Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. 2019. Meta-GNN: On Few-shot Node Classification in Graph Meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.*